Summary statistics from scratch

Group N

Abstract

We present `stats_from_scratch`, a small Python library that implements the mean, the median, the population and sample standard deviations, and the three quartiles directly from their definitions. Every function is checked against the matching function in the standard library `statistics` module; we treat the standard library as a reference implementation in the tests, not in the library code. The library follows the Diataxis framework for documentation [2] and the statistics chapter of the open-source textbook *Python for Mathematics* [1]; we mention the wider statistical context in [3].

1 Introduction

The summary statistics of a sample are the most basic description of its location and spread. For a sample $x_1, x_2, \ldots, x_n$ the arithmetic mean is $\bar{x} = (1/n)\sum_i x_i$, the median is the middle of the sorted sample, and the standard deviation is the square root of the mean squared deviation from the mean. The two common variants of the standard deviation differ in the divisor: n for the population version and $n - 1$ for the sample version [3].

The standard library `statistics` module already implements all of these. We wanted a library that we could read and understand in full, with the formulas in plain sight, and we treat the standard library as the reference implementation in our tests.

We present a five-function library, `stats_from_scratch`; every function is implemented from its definition; and every function is checked against the matching function in the standard library `statistics` module.

2 The library

The library lives in the single module `stats_from_scratch.py` and follows the statistics chapter of *Python for Mathematics* [1].

`mean(values)` returns $\sum_i x_i/n$.

`median(values)` sorts the sample, returns the middle value for an odd count, and the average of the two middle values for an even count.

`population_standard_deviation(values)` and `sample_standard_deviation(values)` return the square root of the mean squared deviation with divisors n and $n - 1$ respectively.

`quartiles(values)` returns the three cut points Q_1, Q_2, Q_3, using the 'exclusive' method of the standard library: the k-th cut is at the fractional 1-indexed position $k(n+1)/4$ of the sorted sample, with linear interpolation between the surrounding values. We picked this method so the library agrees with `statistics.quantiles(data, n=4)` exactly.

3 Worked example

We illustrate the library on the sample $[1, 2, 3, 4, 5]$. The mean is 3, the median is 3, and the three quartiles are $(1.5, 3, 4.5)$, as expected. The population standard deviation is $\sqrt{2} \approx 1.414$ and the sample standard deviation is $\sqrt{2.5} \approx 1.581$. The values match `statistics.mean`, `statistics.median`, `statistics.pstdev`, `statistics.stdev`, and `statistics.quantiles(data, n=4)` to within floating-point precision (Table 1).

Quantity	Library	`statistics`
mean	3.0	3
median	3	3
pstdev	$\sqrt{2}$	$\sqrt{2}$
stdev	$\sqrt{2.5}$	$\sqrt{2.5}$
quartiles	$(1.5, 3, 4.5)$	$[1.5, 3.0, 4.5]$

Table 1: **Agreement with the standard library on** $[1, 2, 3, 4, 5]$**.** Every function in the library returns a value that matches `statistics` to within floating-point precision.

4 Discussion

The standard library already covers this material in full. We wrote the library because we wanted to know what the formulas were doing, and the act of writing them out in clean form was itself instructive.

There are several methods in use [3]. We picked the 'exclusive' method so that the library agrees with the standard library's `statistics.quantiles`. A natural extension would be to expose a "method" keyword so that other methods can be selected.

The library handles only one-dimensional samples. A natural extension is the two-sample case (correlation, covariance, Pearson and Spearman coefficients), each of which has a short definition and an obvious unit-test target in `statistics` or `scipy.stats`.

5 Conclusion

We have presented `stats_from_scratch`, a small library that implements the basic summary statistics from their definitions and agrees with the standard library `statistics` module on every test case we tried.

References

[1] Vincent Knight. Python for Mathematics, 2024.

[2] Daniele Procida. Diátaxis: a systematic framework for technical documentation authoring, 2017.

[3] John A. Rice. *Mathematical Statistics and Data Analysis.* Cengage Learning, 3 edition, 2007.